

Homework #6

Sungwon Cho

1. Radix Sort

Radix sort starts bin sort based on least significant data(i.e last character). So, after first iteration, the list will be sorted by last character and following original order in case of strings which their last characters are same.

Result: **embed, other, refer, class, leaks, every, array**

2. Merge Sort

Right before the last merge, first half array and last half array is already sorted.

Result: **15, 26, 35, 50, 53, 57, 2, 11, 14, 21, 22, 25**

3. Quick Sort

Pivots in this quick sorted picked randomly except 35(given in problem). In each step, number of step means level of depth of recursion.

Step 1-A: Sort list of **35, 57, 53, 26, 50, 15, 22, 21, 25, 14, 11, 2**, pivot **35**

35, 57, 53, 26, 50, 15, 22, 21, 25, 14, 11, 2

35, 2, 53, 26, 50, 15, 22, 21, 25, 14, 11, 57

35, 2, 11, 26, 50, 15, 22, 21, 25, 14, 53, 57

35, 2, 11, 26, 14, 15, 22, 21, 25, 50, 53, 57

25, 2, 11, 26, 14, 15, 22, 21, **35**, 50, 53, 57 (Process Step 2-A, 2-B)

2, 11, 14, 15, 21, 22, 25, 26, 35, 50, 53, 57

Step 2-A: Sort list of **25, 2, 11, 26, 14, 15, 22, 21**, pivot **15**

25, 2, 11, 26, 14, **15**, 22, 21

15, 2, 11, 26, 14, 25, 22, 21

15, 2, 11, 14, 26, 25, 22, 21

2, 11, 14, **15**, 26, 25, 22, 21 (Process Step 3-A, 3-B)

2, 11, 14, 15, 21, 22, 25, 26 (Done)

Step 2-B: Sort list of **50, 53, 57**, pivot **53**

50, **53**, 57

53, 50, 57

50, **53**, 57

50, 53, 57 (Done)

Step 3-A: Sort list of **2, 11, 14**, pivot **11**

2, **11**, 14

11, 2, 14
 2, 11, 14
 2, 11, 14 (Done)

Step 3-B: Sort list of **26, 25, 22, 21**, pivot **25**

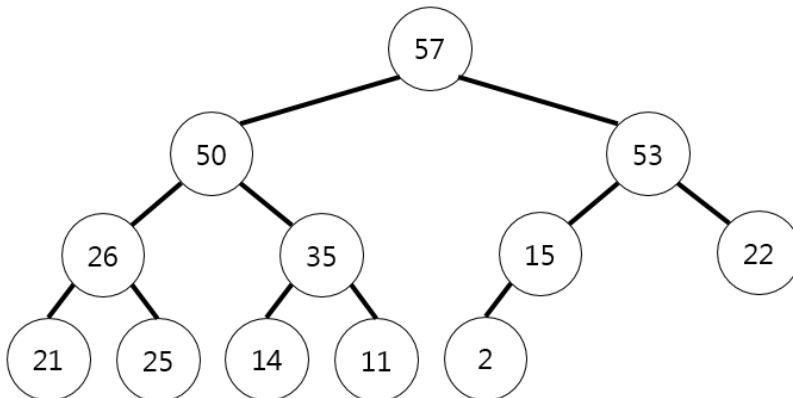
26, **25**, 22, 21
25, 26, 22, 21
25, 21, 22, 26
 21, 22, **25**, 26 (Process Step 4-A)
 21, 22, 25, 26 (Done)

Step 4-A: Sort list of **21, 22**, pivot **21**

21, 22
 21, 22 (Done)

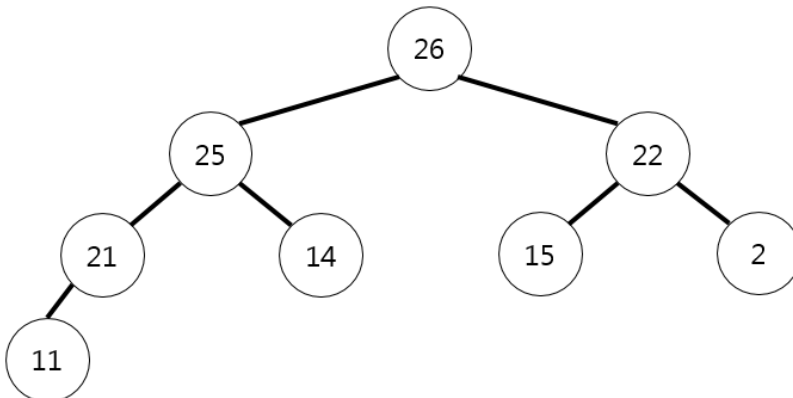
4. Heapify

Just simply make heap by adding elements one-by-one starting from first element of the list.



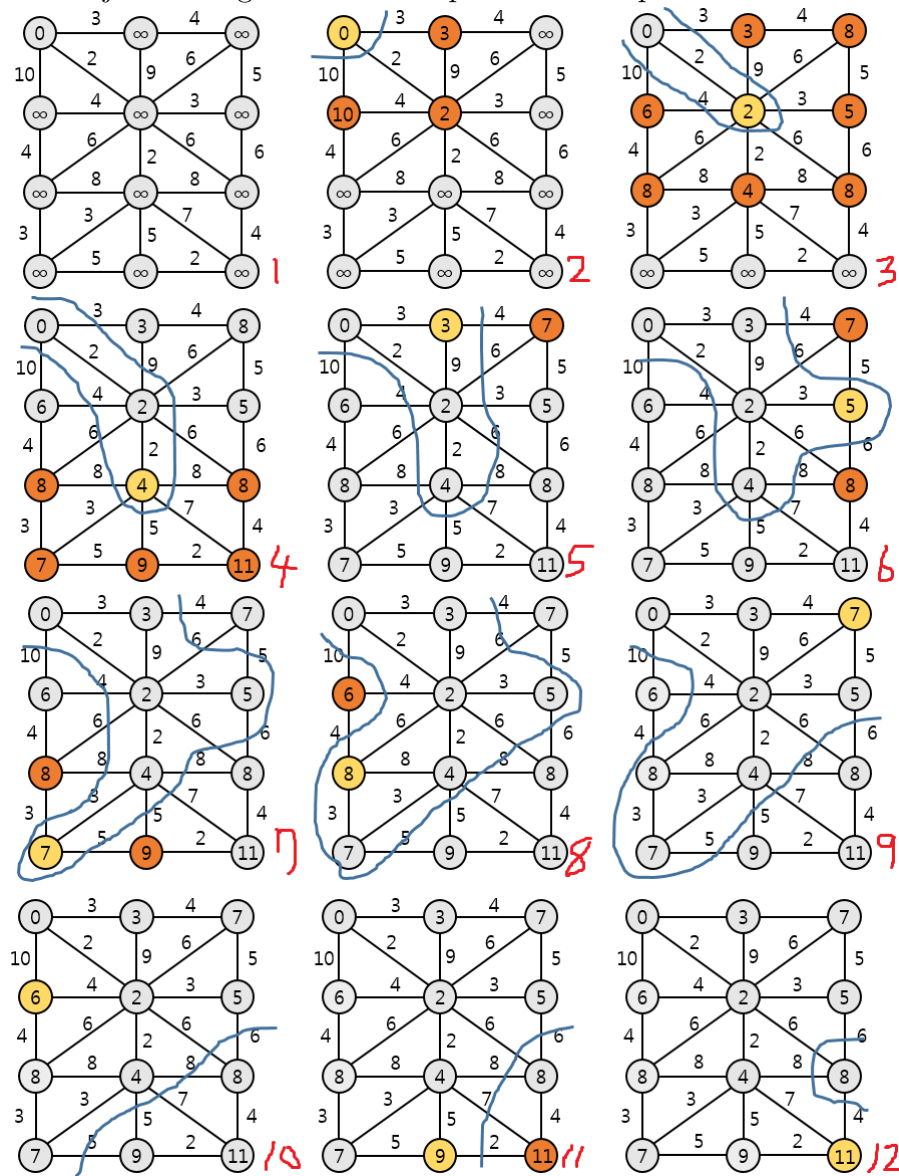
5. Re-heapify

One element is removed for each iteration of heap sort. So, just repeat process of remove one element and re-heapify 4 times to get desired result.



6. Shortest Path

Use Dijkstra's algorithm to compute shortest path from **a** to **k**.



So, length of shortest path from **a** to **k** is 11.